Predicting System Accidents with Model Analysis during Hybrid Simulation

Jane T. Malin
NASA Johnson Space Center
2101 NASA Road 1
Houston, TX 77058
malin@jsc.nasa.gov

Land D. Fleming
Hernandez Engineering, Inc.
17625 El Camino Real, Suite 200
Houston, TX 77058
land.d.fleming1@jsc.nasa.gov

David R. Throop
The Boeing Company
2100 Space Park Drive
Houston, TX 77058
David.r.throop@boeing.com

Keywords: discrete event, flow analysis, static potential, system accident

Abstract

Standard discrete event simulation is commonly used to identify system bottlenecks and starving and blocking conditions in resources and services. The CONFIG hybrid discrete/continuous simulation tool can simulate such conditions in combination with inputs external to the simulation. This provides a means for evaluating the vulnerability to system accidents of a system's design, operating procedures, and control software. System accidents are brought about by complex unexpected interactions among multiple system failures, faulty or misleading sensor data, and inappropriate responses of human operators or software. The flows of resource and product materials play a central role in the hazardous situations that may arise in fluid transport and processing systems. We describe the capabilities of CONFIG for simulation-time linear circuit analysis of fluid flows in the context of model-based hazard analysis. We focus on how CONFIG simulates the static stresses in systems of flow. Unlike other flow-related properties, static stresses (or static potentials) cannot be represented by a set of state equations. The distribution of static stresses is dependent on the specific history of operations performed on a system. We discuss the use of this type of information in hazard analysis of system designs.

BACKGROUND

Simulating Complex System Accidents

System accidents are unsafe states brought about by complex unexpected interactions between failures in systems, and by inappropriate responses to those states. In recent times, there has been increasing interest in predicting and preventing system accidents, as startling system accident cases have accumulated in the years since Perrow described the "normal accident" concept and analyzed the Three Mile Island [1984] and Bhopal [1999] cases, among others. Accidents of this type include an Osprey helicopter crash [Ladkin 2001], Therac-25 failures [Leveson 1995] and the ValueJet 592 crash [Langewiesche 1998]. A key feature of many such accidents is the interaction between an unexpected primary failure and a safety system that was not designed (or maintained in readiness) to handle that failure. Software and instrumentation are often components of an inappropriately designed or maintained safety system that is implicated in an accident. Human operators are also frequently implicated, as they respond to an incomprehensible situation with safety system tools and procedures that are not designed or made ready to handle that situation.

To improve design of complex systems, it is necessary to model new types of "unexpected" and hidden states that have not been previously predicted in analysis and simulation or software testing. In system accidents, failures of components, environment, materials, utilities and instrumentation cause leaks, blocks and incomplete processing that are hazardous. Failures in safety and backup systems can remove barriers to mishaps. Interactions between the primary system inputs and support systems, such as power and thermal systems, can be overlooked. Interaction of latent failures and hazards with operational sequences can be overlooked. Complex configurations that may "look the same" can behave differently, based on hidden built up states and distant influences. For example, local changes in force and flow configurations can cause indirect changes in apparently unconnected components that are in global feedback loops. The key to a system accident is commonly a series of events that leads to a hazardous system configuration that the safety systems and operators are not prepared to handle.

Our goal is to attack this problem by increasing the range of system-level hazards that can be predicted and understood. The first task is to model and simulate event histories that lead to complex hazardous configurations, so that they can be anticipated. Safety and backup systems and procedures can then be designed to appropriately handle these situations. Next, designers can analyze how these complex failures are handled during operations by control, relief or safety systems.

To simulate system accidents, it is necessary to model the forces, resistances and storage that can play a role in complex system events. The CONFIG hybrid discrete/continuous simulation tool, which has these capabilities, has been used for dynamic interactive system-level evaluation of advanced control software [Malin et al. 1998]. In this evaluation, software "requirements errors" were detected in simulated operational scenarios in a complex air processing system for space life support. Unanticipated system configurations were uncovered as the simulation interacted dynamically with the control software.

In this paper, we describe additional CONFIG capabilities that have been developed to model and simulate system accidents. We describe capabilities to globally analyze system configurations and compute flow rates and potentials by methods of graph analysis and linear circuit analysis during simulation. Such analysis includes static potentials generated by flows at points where they are in contact with blockages such as closed valves. The distribution of static potentials is difficult to understand and anticipate because it is dependent on the history of operations performed on the system (e.g., the order in which valve and pumps are operated). In a hydraulic system, an undesirable distribution of static potentials could produce effects such as the unintended opening of a relief valve, with catastrophic consequences.

CONFIG Modeling and Simulation

The CONFIG hybrid simulator extends discrete event simulation with capabilities for approximate and qualitative modeling of continuous system behavior [Malin and Fleming 1999]. This level of fidelity supports investigation of the existence and sequencing of system events in fast scenario-based simulation of operations. When control software is not yet available to interact with the simulation, CONFIG capabilities for scripting and for modeling activities (control, procedures, schedules) can be used for early dynamic analysis of operational problems. CONFIG has been used to model gas and water processing systems, a thermal control system and a data network for space subsystem designers.

Figure 1 shows a graphical representation of one such modeled system, a Variable Configuration Carbon dioxide Removal (VCCR) System, for life support in space. This system is designed to extract CO_2 molecules from the air, for storage in a storage tank (CO2-BUFFER), using sorbent beds with molecular sieve technology. The system configuration is cycled periodically, to alternately fill or empty one of two sorbent beds, B3 and B4. B1 and B2 are used for humidity control. This screen capture shows the graphical user interface for the VCCR system operating during simulation. The arrows superimposed on connections indicate directions of gas flows in a situation where B3 is desorbing and B4 is adsorbing. The VCCR-CTL oval (in HC1-B3-Desorb state) is an activity that models software control of the VCCR duty cycle.

The discrete event simulation base provides a framework for causal modeling of states and outcomes, and specifying transition functions that are internal or triggered by external inputs. The discrete event system specification formalism introduced by Zeigler [1976] specifies a system with a continuous time base, discrete inputs and state transitions. The continuous time base of discrete event simulation supports both event-stepped time advances and

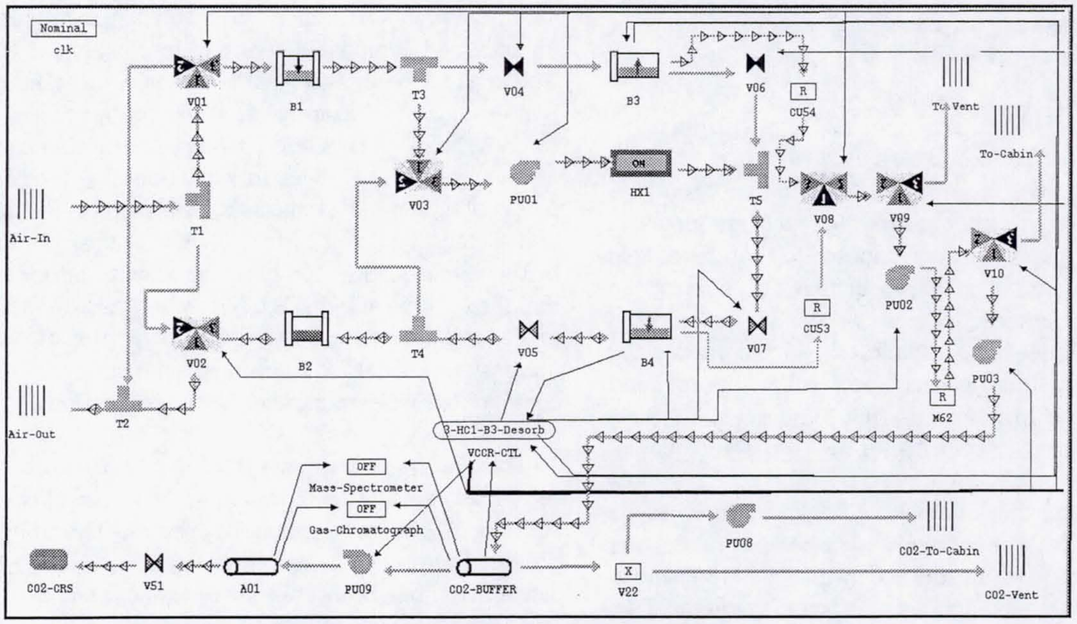

Figure 1. CONFIG model of a Carbon Dioxide Removal System

discrete-time-step approaches to continuous simulation. In CONFIG, a discrete-time-step approach, with either linear or exponential approximation [Throop et al. 2000], can be used to periodically update continuously changing variables in a component.

Discrete event models of systems are composed of coupled interacting component models. System behavior emerges from the coupled behavior of the components. In CONFIG, the model structure can be "recomposed" during a simulation as the direction and activation of the couplings changes. Parts of the model are activated or deactivated as operating modes of components change or closed off areas of the system are brought into the working system configuration. Recently, the coupled model approach has been used in a capability for selection and mixing of simple and complex subsystem models in simulation experiments. Selecting versions of subsystem models can focus and speed up simulations.

CONFIG simulates complex flow regimes – multi-component mixtures, mixed-phase flows, variations in pressure, temperature and fluid density. A global flow and pressure/potential analysis capability, the Flow-Path Management Module (FPMM) tracks dynamic changes in system configurations and model structure during simulation [Malin and Fleming 1998]. The same underlying facility supports computation for both fluid and electrical flow at the abstracted level of effort, flow and resistance. In each simulation step, both local and global calculations produce the system behavior. Locally, the component behavior is calculated from its inputs and its state. Externally triggered and internally driven transitions result in changes in time-delayed value assignments.

In CONFIG, components have multiple behavior modes. Each mode has state equations that generate behavioral effects, and conditions that govern mode transitions. CONFIG activity models support modeling of controllers, human operator procedures, actions and schedules. Object-oriented model types for devices support modeling of components. In discrete event simulation, generation of events and time-advances can be random, supporting probabilistic analysis. In use thus far, CONFIG simulations have been deterministic, supporting analysis of specific state configurations and inputs. The control software interacts with and manages the simulated hardware.

CONFIG models and simulations can represent the effects of failures and other problems. CONFIG provides capabilities to model failures of configuration, input, capacity, performance, control and operations. Certain types of failures and problem inputs may trigger discrete changes in behavior. These can change the control regime, the system configuration, or the capacity or performance of a system component. For example, discrete changes can result from bursts, shorts, errors or uncommanded actions. Changes in control, capacity or performance can also be continuous, gradual and nonlinear. These types of changes can result from buildups, wear, leaks and drifts. Some failures result in component states that cannot change when they should. Examples are stuck components. Some failures involve random variation in measurements or inputs. For any failure, the magnitude of the effect may be determined by the magnitude of the failure. Component failures can produce problem inputs elsewhere in the system. These result in cascades of failures and off-nominal component states. In complex systems, these cascades can be difficult to anticipate during design.

SYSTEM FLOWS AND POTENTIALS

The CONFIG Flow-Path Management Module (FPMM) supports simulation of fluid flows and the potentials associated with them. When a CONFIG model is constructed, a system of linear equations is generated in the background for the directed graph of components and connections. The equations relate the resistances and sources of effort to flows and potentials according to Ohm's law for linear circuits and Kirchhoff's Laws for voltage and current. (To generalize from the electrical domain to the larger domain of fluid flows, we refer to "potentials" rather than the more specialized term "voltages.") From the perspective of the FPMM, resistances and efforts determine flows and potentials, which are thus the dependent variables. Both resistances and efforts may vary during the course of a simulation, with the resultant changes in rates of flow and potentials computed and set.

Ohm's Law is expressed by the equatio n: $\Delta P=FR$, where ΔP is the potential across a two-terminal resistance element of a circuit, F is the flow (electrical or fluid) across the terminals and R is the resistance of the element. Kirchhoff's Voltage Law states that the sum of the potential drops across circuit elements around any loop is zero. The current law states that the sum of currents leaving any circuit element is zero.

Using the CONFIG device description language, "process statements" may be written for fluid storage devices such as tanks, where changes of internal tank pressures depend on rates of fluid flow into or out of the tank. Because internal tank pressures are sources of effort on the paths of flow over which fluids are entering or leaving a tank, changes of pressure computed by such process statements in turn influence the rates of flow in the system computed by the FPMM. Similarly, a control device such as a pressure regulator or flow controller may be modeled so that its resistance varies in response to changes in potential or flow across the device, thus also influencing the rates of flow computed by the FPMM. Thus, from the standpoint of the CONFIG process language, flows and potentials determine resistances and efforts. From the standpoint of the FPMM the relationship is reversed, and resistances and efforts determine flows and potentials.

While the linear relationship of effort to flow of Ohm's law is inexact for fluids, the resultant qualitative behavior of simulated fluid system models has proven useful for providing insight into system design and operation and for interactive testing of software that controls the operations of a modeled system.

The utility of the FPMM capabilities to hazard analysis are obvious: toxic fluids flowing through a system may pose a direct hazard, while off-nominal flows of even nontoxic fluids such as water used for coolant may contribute to hazardous situations. Of course, the importance of such capabilities to useful simulations has long been recognized and CONFIG is far from the only simulation tool in which representations of flow have been incorporated in one form or another. However, one important aspect of fluid systems has for the most part been missing in simulation software: a method for representing the static stresses induced in the modeled system by the same sources of effort responsible for generating fluid flows.

Perrow [1984] describes a mishap at a small nuclear reactor at Humboldt Bay, California that was exacerbated because the operators "... assumed that a safety valve had opened to reduce pressure [in the reactor core]. But instead, a different safety valve opened, and, due to coolant shrink from its discharge, a low-water level signal came on." A lack of knowledge about the state of static potentials in the system played a role in this accident.

Static stresses are measured in the same units as are potential drops across resistive elements carrying flow. The two properties are closely related but quite distinct from each other; a nonzero static stress may be thought of as a potential drop across an infinite resistance, but Ohm's Law gives no solution because the result of multiplying a zero flow by an infinite resistance is indeterminate. To emphasize both the relationship and differences of the two properties, we will henceforth refer to static stresses as "static potentials" and refer to potentials across flow-conducting elements of finite resistance as "dynamic potentials."

For purposes of computation, dynamic potentials can be considered state variables because their values are independent of the sequence of events that led to the present state of the system. Given the set of efforts, resistances, and the network topology describing a circuit, all dynamic potentials are completely determined. This is illustrated in Figure 2.

The potential drops across the three finite resistances, R_a, R_b, and R_c, are fully determined by the resistance and effort values in accordance with Ohm's Law and Kirchhoff's Law. The potential drop, P_a across resistor R_a is: $P_a = ER_a/(R_a + R_b)$. The potential drop across resistance R_b is $P_b = ER_b/(R_a + R_b)$.

Because of the series of open switches S_1, S_2, and S_3, there is no flow through resistance R_c, so the potential across that element is zero.

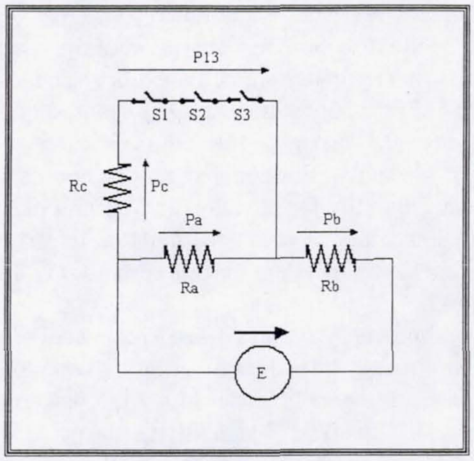

Figure 2. A simple electric circuit

Kirchhoff's Voltage Law requires that the potential P_{13} across the entire series of three switches must be equal to P_a. However, the resistances and effort alone cannot determine the potential across any individual switch of the series. These potentials are determined by the history of configuration actions performed on the system and changes in magnitudes of the resistances or effort. Two possible scenarios leading to the final state of the circuit in Figure 2 illustrate the dependency of static potentials upon the specific trajectory through intermediate states to a final state. The alternative scenarios are shown in Figure 3 and

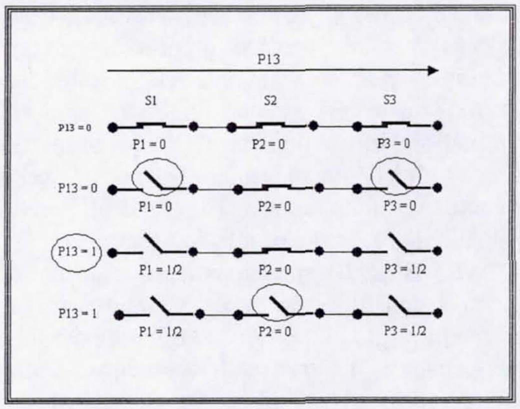

Figure 3. One possible scenario leading to the circuit configuration of Figure 2

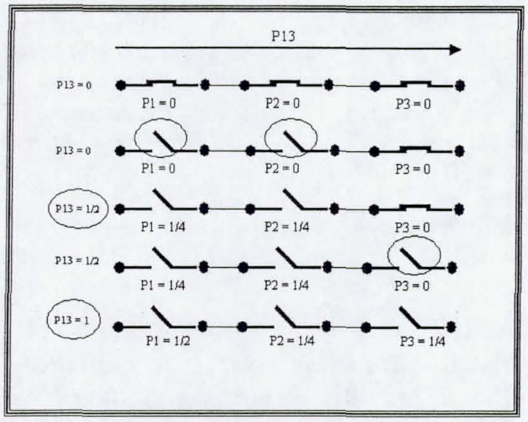

Figure 4. An alternative scenario for achieving the circuit configuration of Figure 2

Figure 4. In the initial configuration for both scenarios, all switches are closed and the value of the effort source E is zero, so that all potentials are also zero, according to Ohm's Law. The operations that produce each subsequent state are indicated by ellipses.

For states achieved by changing the applied effort, E, we assume the switches are identical, so that the two switches that are connected to the "live" portion of the circuit are each assigned half of the resultant increment to P_{13}. At the end state of the first scenario, the potential across S2 is 0 and switches S1 and S2 both have potentials of ½. In the second scenario, the effort is set so that P_{13} is ½ in an intermediate state and then 1 in the final state, leaving a potential of ¼ "trapped" across switch S2. The final values of P_{13} and of all the dynamic potentials are the same in the end states of both scenarios.

With a continuously variable source of effort and, perhaps, continuously variable resistances, even for a very simple circuit such as this the number of sets of possible values for the static potentials across the three switches is clearly infinite.

High-voltage electrical systems could easily be envisioned for which many different sequences of operations lead to the same nominal distributions of currents and dynamic potentials in the circuit with some sequences resulting in unexpectedly high static potentials at points in the circuit. The unexpectedly high potentials could lead to arcing. In an environment containing explosive gases, this clearly could be extremely hazardous.

In fluid processing and transport systems of the sort that CONFIG has been used to simulate, a seemingly reasonable sequence of operations could result in the leakage of valves stressed beyond design limits or the unexpected opening of relief valves, potentially releasing toxic chemicals.

Computing Static Potentials

For some of the states of the circuit in our illustration, we distributed the numerical values of static potentials among the three switches based on the assumption that these devices are "identical" in some way.

The relevant property is capacitance, *C*, which in the electrical domain is defined by the equation: $q = C\Delta P$, where q is the charge stored in a two-terminal circuit element and ΔP is the potential difference between terminals. In the fluid flow domain, stored mass is analogous to electrical charge.

The CONFIG Flow-Path Management Module sets the static potentials for any circuit element at which flow (electrical or fluid) is blocked according to the constraints imposed by Kirchhoff's Voltage Law and by a conservation law. For electrical circuits, charge must be conserved and for fluid circuits, fluid mass must be conserved.

FPMM partitions the circuit into areas such that each area, *A*, consists of a set of interconnected unblocked elements and is bounded by a set of *n* blocked circuit elements, each connected to one or more elements internal to *A*. An expression derived from Kirchhoff's Law and the conservation law is then used to compute the static potential on each blocked element induced by the flows and efforts within the area's conducting elements. The static potential at each blocked element *j* that bounds area *A* is:

$$\Delta P_j = \sum_{i \neq j}^{n} C_i V_{ij} \Big/ \sum_{k=1}^{n} C_k$$

where V_{ij} is the dynamic potential difference on any path within the area *A* traversed from blocked element *i* to blocked element *j*, treating all potentials external to area *A* as zero. The total potential across any blocked element in the circuit is then the sum of the potentials induced in the element by the two adjoining areas that the element separates.

In the present implementation of the FPMM, capacitances are assumed to be unity for all circuit elements having an infinite resistance (e.g., open switches and closed valves). This simplification is sufficient for the semi-qualitative purposes of CONFIG simulations of fluid system designs, where detailed information on device capacitances is generally unavailable. During simulations, the FPMM static potential utility maintains information on the associations between blocked and unblocked circuit elements and incrementally updates static potentials when resistances or efforts are changed. When valves or switches transition between blocked and unblocked states, the association data is updated and static potentials are recomputed as is appropriate.

The static potential utility has been used with several modeled systems, including the VCCR model of Figure 1. For the three-way valves shown, each black triangle represents a blocked port. It can be seen that some of these blocked ports separate paths of active flow, such as port#2 of Valve V03. Determining the correct magnitudes of the static potential across blocked ports in such a complex system can be a laborious process for a human, especially in a system such as the VCCR where valve operations and the associated flow redirections are part of the normal operating cycle. The FPMM, however, can update the potentials for each reconfiguration event with only a small proportion of simulation CPU time.

CONCLUSIONS AND FUTURE WORK

With the computation of static potentials, we achieve a new level of understanding of indirect effects in complex systems. The capability to alternate global analysis of flow paths with local computation of events during simulation supports predicting indirect effects. Tracking static potentials makes it possible to simulate the effects of history as well as state during reconfigurations. These global and history-based effects are some of the most difficult for designers to anticipate and for operators to understand.

Recently, we have begun collaborating with developers of the Brahms tool for modeling human work and activities [Clancey et al. 1998]. Integrated use of these tools can enhance analysis of operational aspects of designs. We are beginning a project that will use CONFIG in a system to support model-based hazard analysis for complex systems.

REFERENCES

Clancey, W. J.; P. Sachs; M. Sierhuis; R. van Hoof. 1998. "Brahms: simulating practice for work systems design." *International Journal of Human-Computer Studies 49*: 831-865.

Ladkin, P. 2001. "Software Direct Cause of December 2000 Osprey Crash." *The Risk Digest 21,* Issue 33, April.

Langewiesche, W. 1998. "The Lessons of ValueJet 592." *The Atlantic Monthly*, March 15: 81-98.

Leveson, N. 1995. *Safeware: System Safety and Computers*. Reading, Mass.: Addison-Wesley.

Malin, J. T. and L. D. Fleming. 1998. "Global Qualitative Flow-path Modeling for Local State Determination in Simulation and Analysis". U.S. Patent 5,732,192.

Malin, J. T., and L. Fleming. 1999. "Enhancing Discrete Event Simulation by Integrating Continuous Models." In *Hybrid Systems and AI*. Working Notes for AAAI 1999 Spring Symposium Series (Stanford, CA, March 22-24). AAAI, Menlo Park, CA.

Malin, J. T.; L. Fleming; T. R. Hatfield. 1998. "Interactive Simulation-Based Testing of Product Gas Transfer Integrated Monitoring and Control Software for the Lunar Mars Life Support Phase III Test." In *Proceedings of SAE 28th International Conference on Environmental Systems*. SAE Paper No. 981769.

Perrow, C. 1984 & 1999. *Normal Accidents: Living with High Risk Technologies*. Princeton Univ. Press, Princeton, NJ.

Throop, D. R.; J. T. Malin; L. Fleming. 2001. "Automated Incremental Design FMEA." In *IEEE Aerospace Conference Proceedings 2001*, (CD). IEEE Aerospace Conferences, Manhattan Beach CA.

Zeigler, B. P. 1976. *Theory of Modeling and Simulation*. New York: Wiley.

BIOGRAPHY

Jane T. Malin is Technical Assistant in the Intelligent Systems Branch in the Automation, Robotics and Simulation Division in the Engineering Directorate at NASA Johnson Space Center, where she has led artificial intelligence research projects for 17 years. She has led development of the CONFIG simulation tool for evaluating intelligent software for operation of space systems. She has led research on intelligent user interface and intelligent systems for control and fault management of space systems. Her 1973 Ph.D. in Experimental Psychology is from the University of Michigan.

Land D. Fleming is a Computer Systems Specialist supporting the NASA Johnson Space Center Automation, Robotics, and Simulation Division since 1990. He has been involved in both the development of computer simulation tools and their application to space systems. His 1987 M. S. in Computer Science is from De Paul University.

David R. Throop has been an Artificial Intelligence Specialist with The Boeing Company since 1992. He provides engineering software support in the Intelligent Systems Branch in the Automation, Robotics and Simulation Division in the Engineering Directorate at NASA Johnson Space Center. He oversaw development of FMEA modeling software and its use for the International Space Station. His 1992 Ph.D. in Computer Science is from the University of Texas, with a dissertation on Model Based Diagnosis. His 1979 Bachelors of Chemical Engineering is from Georgia Tech.